

バースト誤りと輻輳の双方を考慮した アドホックネットワークにおける TCP 通信高速化方式の検討

春原 真人 糸川 一也 加藤 聰彦 伊藤 秀一

電気通信大学 大学院情報システム学研究所

〒182-8585 東京都調布市調布ヶ丘 1-5-1

E-mail: {suno, kumekawa, kato, itoh}@net.is.uec.ac.jp

無線リンクのみでネットワークを構成するアドホックネットワークにおいて TCP 通信を行う場合には、伝送誤りやノードの移動に伴うリンクの切断によるパケットロスにより不要な輻輳制御が起動され、スループットが低下するという問題が発生することがある。このような問題を解決するために従来から多くの無線 TCP の手法が提案されているが、これらの手法は無線リンクでは伝送誤りやリンクの切断のみが生じ、輻輳は生じないことを前提としている。しかし、無線リンクのみでネットワークを構成するアドホックネットワークにおいては、これらに加えて無線リンクにおける輻輳を考慮した方法を導入する必要があると考えられる。そこで本稿では、無線アドホックネットワークにおいてバースト誤りと輻輳の双方を考慮した TCP 通信高速化方式について述べる。

High Speed TCP Scheme Considering Both Burst Errors and Congestions over Wireless Ad hoc Networks

Masato SUNOHARA Kazuya KUMEKAWA Toshihiko KATO Shuichi ITOH

Graduate School of Information Systems, University of Electro-Communications

1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan

E-mail: {suno, kumekawa, kato, itoh}@net.is.uec.ac.jp

In the ad hoc networks structured with only wireless links, the packet losses due to the transmission errors or link down resulting from node movement may reduce the TCP throughput due to the unnecessary invocations of congestion control. Although there have been proposed several wireless TCP mechanisms to solve this problem, they assume that a wireless link is used as the last hop link in the network and only transmission errors and link downs occur over the wireless link. However, the congestion over wireless link needs to be also considered in the ad hoc networks constructed by only wireless links. In this paper, we propose a high speed TCP scheme considering both the burst errors and congestions over wireless ad hoc networks.

1. はじめに

近年アドホックネットワークが注目され、さまざまな応用が検討されている。信頼性の高いデータ転送を必要とするアプリケーションでは、通常のインターネットと同様に TCP が使用される。アドホックネットワークでは無線リンクを使用する機会が多いため、伝送誤りや、ノードの移動

に伴うリンクの切断によりパケットロスが発生し、TCP 通信において不要な輻輳制御が起動されスループットが低下するという問題が発生することがある。このような無線 TCP に対しては、従来から多くの技法が提案されている[1]。しかし、従来の技法は最後の 1 ホップに適用されることを想定し、無線リンクでは伝送誤りまたはハンドオフのみが生じ、輻輳は発生しないことを前提

としている。これに対し無線リンクのみでネットワークを構成するアドホックネットワークでは、特定のノードにトラフィックが集中した場合は、輻輳が発生する可能性がある。すなわち、伝送誤り、移動に伴うリンクの切断、輻輳のすべてを考慮した方法を導入する必要があると考えられる。そこで本稿では、バースト誤りと輻輳の双方を考慮した TCP 通信高速化方式について述べる。2 章で本方式の概要について、3 章で詳細設計について示し、4 章で本方式をネットワークシミュレータ ns-2 上に実装し評価した結果について述べる。

2. 方式の概要

2.1. アドホック TCP の振舞い

筆者らは、AODV[2]と 802.11 無線 LAN を用いたアドホックネットワークにおいて、TCP トラフィックが特定のノードに集中した場合どのように輻輳が発生するかをシミュレーションにより評価した[3]。その結果、ノード内のバッファオーバーフローは少なく、RTS の再送がリトライアウトしたことによるパケットロスが多発した。RTS の再送がリトライアウトした原因は、RTS の受信ノードにおいて RTS のリンク上の衝突が起きたことによるものか、受信ノードに RTS が届いているにもかかわらず受信ノードが他のノードと通信中であるため CTS を返さないことによるものであった。すなわち、無線チャネルの競合によるパケットロスが多発するという結果であった。また同様な通信状況においても評価した[4]。ランダムなビット誤りは 802.11 無線 LAN の再送により回復可能なため、この評価ではバースト的にビット誤りを発生させた。その結果、データパケットの再送がリトライアウトしたことによるパケットロスと、RTS の再送がリトライアウトしたことによるパケットロスが多発し、輻輳制御が起動されスループットが大きく低下した。この結果において、無線チャネルの競合が原因となる RTS のリトライアウトによるパケットロスは、ネットワークの輻輳と判断すべきであるが、データパケットのリトライアウトやビット誤りが原因となる RTS のリトライアウトによるパケットロスに対しては、輻輳と判断するのではなく、バースト誤りに適する処理を起動すべきであると考えられる。そこで、以下のアプローチを採用することとした。

2.2. パケットロスの原因の通知

上述のようにパケットロスの原因によって異なる処理を行うためには、パケットロスの原因を送信ノードに明示的に通知する必要がある。伝送誤りを通知するものとして ELN (Explicit Loss Notification)[5]、EBSN (Explicit Bad State Notification)[6]があり、これにより伝送誤りによる再送が生じて、輻輳制御を起動しない処理が可能となる。また移動によるリンクの切断を通知するものとして SCPS-TP[7]、TCP-F[8]、ELFN (Explicit Link Failure Notification)[9,10]などがあり、リンクの切断に対しては送信を中止しリンクの再確立を待って送信を再開することができる。輻輳の明示的な通知には ECN (Explicit Congestion Notification)[11]があり、輻輳の通知時のみに輻輳制御を起動することが可能である。本方式では、これらを参考にして、以下の手順によってパケットロスの原因を送信ノードに明示的に通知する。図 1 にその概念図を示す。

- (1) パケットの中継ノードは MAC 層の情報を監視し、パケットロスの原因が輻輳またはバースト誤りであると判断し、その情報を次に到着した同一 TCP コネクションのデータパケットに、フラグとして付加し転送する。
- (2) TCP 受信ノードは、(1)で付加されたフラグ付きの TCP のデータパケットから、パケットロスの原因を判断し、その情報を ACK パケットにフラグとして付加し送信する。

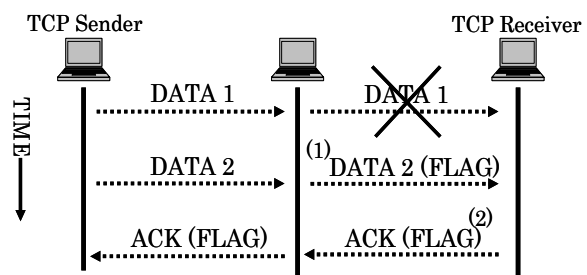


図 1 パケットロスの原因通知の概念図

2.3. バースト誤りと輻輳に対応する処理

上記の手順によりバースト誤りまたは輻輳の明示的な通知を受けた TCP 送信ノードは、それぞれの場合において次の処理を行うこととする。

- バースト誤り

3つの重複ACKを受信しても輻輳制御を起動しない。その一方で、以下のように、SACK オプ

ションの情報を用いて積極的に再送を行う。この再送は、従来の規定[12]とは異なり、SACK オプションで選択的に受信応答がなされていないセグメントを再送することを目的としている。

- 輻輳

通常どおり輻輳制御は起動し、SACK オプションによる再送は行わない。

3. 詳細設計

3.1. 輻輳とバースト誤りを通知するビット

TCP セグメントを中継するノードが、パケットロスの原因の情報を付加するために、以下のようなフラグを導入する。

- 輻輳かバースト誤りのどちらかにより、RTS がリトライアウトしたことを通知するための RTS フラグ
- バースト誤りによるパケットロスが発生したことを通知するビット BE (Burst Error Experience)フラグ
- 輻輳によるパケットロスが発生したことを示すビット CE (Congestion Experience)フラグ

また TCP 受信ノードが、中継ノードから通知されたパケットロスの原因の情報を、TCP 送信ノードに通知するために、以下の 2 つのフラグを導入する。バースト誤りによるパケットロスが発生したことを示すビット EBN (Explicit Burst Error Notification)フラグと、輻輳によるパケットロスが発生したことを示すビット ECN (Explicit Congestion Notification)フラグである。

3.2. 中継ノードの振舞い

RTS のリトライアウトやデータフレームのリトライアウトは、MAC フレームの送信ノードで検出可能である。また、チャネルの競合(輻輳)が原因で RTS のリトライアウトが生じたことは、MAC フレームの受信ノードでのみ検出可能である。このため、RTS フラグと BE フラグの設定はフレームの送信時に行い、RTS フラグが設定された場合の原因の調査は、MAC フレームの受信ノードで行う。詳細な手順を以下に記述する。

各ノードは自身が中継する TCP 通信を監視し、TCP コネクション毎に、以下の情報を保持するテーブル(コネクションテーブル)を管理する。

- (1) バースト誤りの発生
- (2) バースト誤りまたは輻輳の発生

- (3) フレーム衝突またはチャネルビジーの発生

各ノードは、TCP データセグメントを含むフレームのリトライアウトが発生した場合には、バースト誤りによってパケットロスが生じたと判断し、対応する TCP コネクションのテーブルの(1)の情報を TRUE にする。また RTS のリトライアウトが発生した場合には、バースト誤りまたは輻輳によるパケットロスが起きたと判断し対応するテーブルの(2)の情報を TRUE にする。一方、フレームの受信時に衝突が発生したか、または他のノードと通信中であるため受信した RTS に対する CTS を返さなかった場合には、輻輳が起きていると判断し、すべてのコネクションテーブルの(3)の情報を TRUE にする。

各ノードは TCP セグメントを含むデータフレームを受信すると、コネクションテーブルをチェックし、以下の処理を行う。

- (1)の情報が TRUE だった場合
パケットに BE フラグをたてて転送する。
- (2)の情報が TRUE だった場合
パケットに RTS フラグをたてて転送する。
- パケットに RTS フラグがたっていた場合
(3)の情報が TRUE であれば、RTS フラグがついた理由が輻輳であると判断し RTS フラグを消去して CE フラグをたてて転送する。また TRUE でなければ RTS のリトライアウトはバースト誤りにより生じたと判断し、BE フラグをたてて送信する。なお、RTS フラグはクリアする。

前者の 2 つの場合は、この後対応する情報をクリアする。また(3)の情報に対しては、発生後一定時間後にクリアする。

3.3. TCP 受信ノードの振舞い

TCP 受信ノードは、中継ノードによってパケットに付加されたフラグの情報を管理するリスト(フラグリスト)を、コネクション対応に保持する。リストの一つのエントリは以下の 3 つの情報を管理する。

- シーケンス番号
- BE フラグ
- CE フラグ

BE フラグまたは CE フラグ付きのデータパケットを受信すると、そのパケットのシーケンス番号と BE フラグまたは CE フラグの情報を含むエントリをフラグリストに追加する。対応する ACK パケットを送信する際には以下の処理を行う。

- (1) ACK パケットの確認応答番号より古いシーケンス番号のエントリはフラグリストから削除する。
- (2) フラグリストをチェックし、BE フラグ付きのエントリがあり、CE フラグ付きのエントリが無いならば、ACK パケットに EBN フラグをたてて転送する。
- (3) CE フラグのエントリがあり BE フラグのエントリが無い場合には、ACK パケットに ECN フラグをたてて転送する
- (4) BE フラグと CE フラグが共にある場合にはフラグはたてずに転送する。

このようなACKパケットにはSACKオプションも付加される。

3.4. TCP 送信ノードの振舞い

TCP 送信ノードは、EBN フラグと ECN フラグを持つ ACK パケットに対して、次のような処理を行う。

- EBN フラグ

重複 ACK カウントの増加を行わないことによって、重複 ACK が 3 つ届いたときの輻輳制御を回避する。また SACK オプションを用いた積極的な再送を行う。

- ECN フラグ

輻輳が発生したと判断し、SACK オプションによる積極的な再送は行わない。

SACK オプションを用いた積極的再送の手順は以下のとおりである。TCP 送信ノードは送信したが未だ確認応答されていないパケットのシーケンス番号の情報を管理するリストを持つ。新しいパケットを送信すると、そのパケットをリストの最後尾に付ける。そして、ACK パケットを受信すると、以下の手順で処理を行う。

- (1) ACK パケットの確認応答番号よりも古いシーケンス番号のエントリを削除する。
- (2) SACK オプションがあれば、SACK ブロックに指定されたシーケンス番号のエントリに SACK フラグをたてる。
- (3) リスト内の SACK フラグをたてたエントリの中で、リストの最後尾に一番近いエントリよりも前にある SACK されていないエントリのシーケンス番号のパケットを再送する。
- (4) 再送したパケットのシーケンス番号を持つエントリをリストの最後尾に移動する。

手順(4)の処理により、一度再送したパケットが

ロスした場合も再度送信することができる。

4. 性能評価

4.1. シミュレーション条件

シミュレーションには ns-2 を使い、バースト誤りが生じている場合の評価として、距離が 90m 離れた二つのノード間の 1 本の TCP 通信でバースト誤りが生じている状況とし、また輻輳が生じている場合の評価として、図 2 に示すネットワーク構成でノード 0、1、7、10 からノード 4 宛での 4 本の TCP 通信とした。ルーティングプロトコルとしては AODV (Ad hoc On-demand Distance Vector) を、データリンクレベルとして IEEE802.11 無線 LAN を用いる。シミュレーション条件は、伝送速度：11Mbps、ノード移動：なし、仮想キャリアセンス：あり、AODV のリンク切断検出：Hello パケット、セグメントサイズ：1000 バイト、ウィンドウサイズ：20 セグメント、ノードのバッファ：500 パケットである。バースト誤りとしては、パケットエラー有り と パケットエラー無し の 2 状態マルコフモデルを用いた。マルコフモデルではパケットエラー有りの状態から無しの状態に遷移する確率を 0.8、無しの状態から有りの状態に遷移する確率を 0.2 とし、各状態を 1 秒間継続することとした。

4.2. バースト誤りを対象とした評価

パケットエラーレートを 0.0 から 1.0 まで変化させて、TCP と本方式のスループットを比較した結果を図 3 に示す。またパケットエラーレート 0.5 の場合のシーケンス番号の変化を図 4 に、輻輳ウィンドウの変化を図 5 に示す。従来の TCP はエラーレート 0.5 で急激にスループットが低下しているのに対して、本方式はスループットの

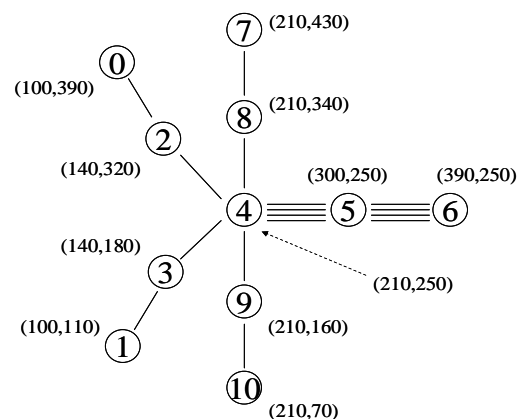


図 2 ネットワーク構成

低下を防いでいることがわかる。また図5から、従来のTCPに比べて本方式では、輻輳ウィンドウが高い位置を保っていることから不要な輻輳制御を防いでいることがわかる。エラーレートが0.4以下の場合に性能に差が出ていない原因は、転送率を上げて、帯域に限界が有りこれ以上性能が上がらないためと考えられる。

表1にエラー0.5の場合の packet loss の内容を示す。packet loss の原因は、RTS のリトライアウトまたはデータパケットのリトライアウトであった。従来のTCPでは、このようなpacket loss に対して、不要な輻輳制御を起動し、また再送は3回の重複ACKを受信したときの高速再転送かタイムアウト再送のみである。これに対して、本方式では、不要な輻輳制御を起動せず、またSACK オプションを用いた積極的な再送しているため、スループット低下を防ぐことができたと考えられる。

4.3. 輻輳を対象とした評価

従来のTCPと、本方式と、輻輳を考慮しない方式とを比較する。輻輳を考慮しない方式とは、本方式で前述の輻輳を示すECNフラグをたてないものである。図6に従来のTCPと本方式のシーケンス番号の変化を示す。本方式では、輻輳によるpacket loss に対しては従来のTCPと同じように輻輳制御を起動することとしているので同じグラフになっている。図2は輻輳を考慮しない方式である。この方式では、輻輳によるpacket loss に対してもSACK オプションを用いて積極的に再送を行い、さらに3回重複ACKが来たときの輻輳制御を起動しないため、TCPの送信ノードでスループットが偏っていることがわかる。従って、輻輳によるpacket loss に対しては

表1 パケットロスの内容

		RET (RTS)	RET (DATA)
従来のTCP	データ	129	177
	ACK	104	47
	合計	233	224
提案方式	データ	181	212
	ACK	151	93
	合計	332	305

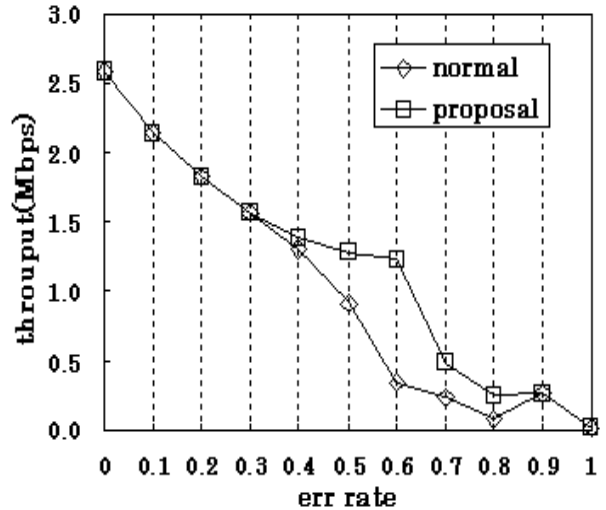


図3 スループット

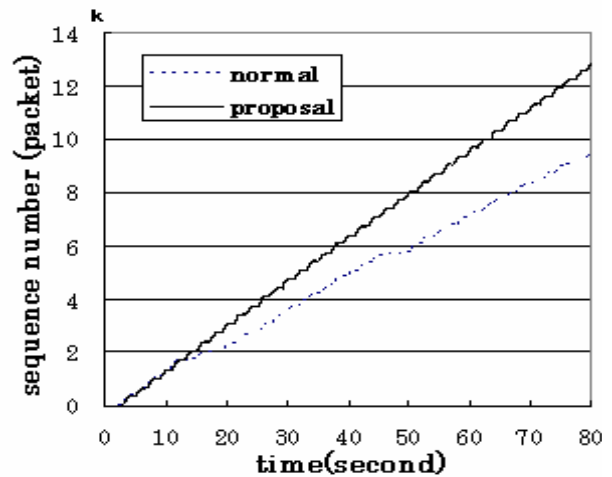


図4 送信シーケンス番号の時間変化

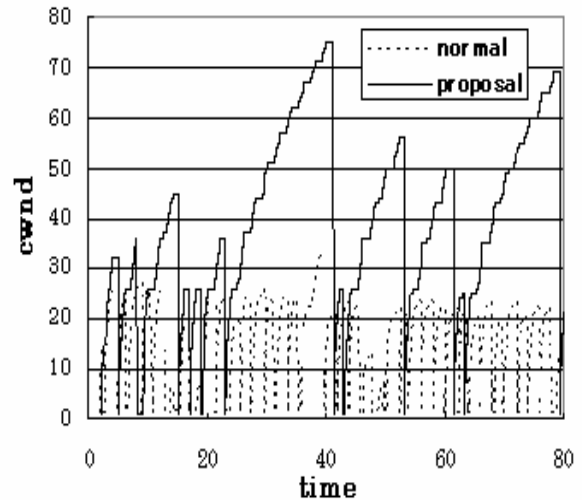


図5 輻輳ウィンドウの時間変化

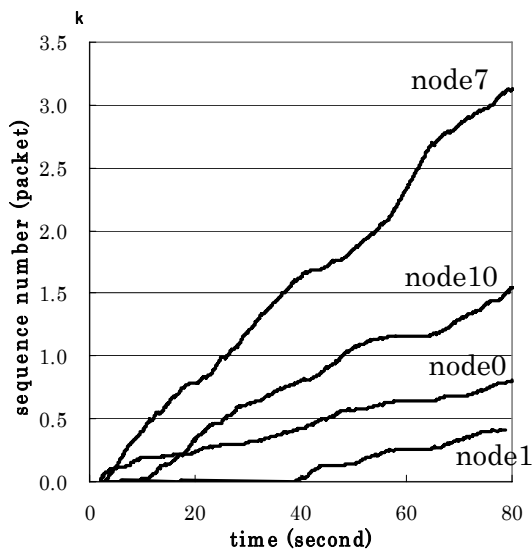


図6 通常のTCPと本方式のシーケンス番号の時間変化 (コネクション別)

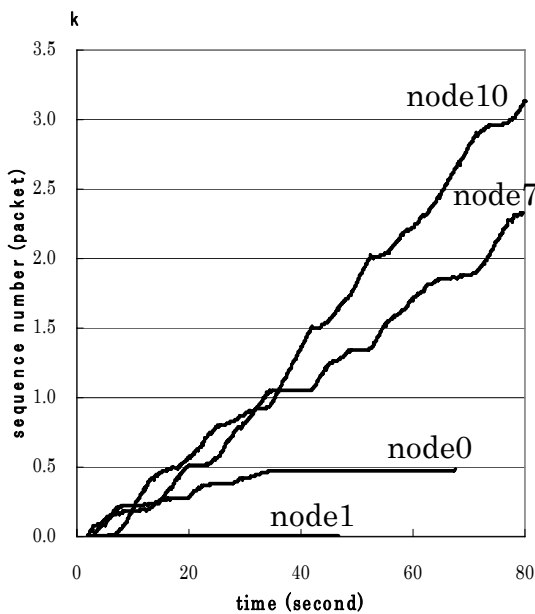


図7 輻輳を考慮しない方式のシーケンス番号の時間変化 (コネクション別)

輻輳制御を起動する必要があり、本方式ではそれが考慮されていることがわかる。

5. おわりに

本稿では、バースト誤りと輻輳の双方を考慮したアドホックネットワークにおけるTCP通信高速化方式の概要と詳細設計と、ns-2を用いて性能評価を行った結果を示した。バースト誤りを対

象とした評価では、本方式は従来のTCPより性能が上がり、また輻輳を対象とした評価では、従来通りの輻輳制御を起動することができる事が明らかとなった。

参考文献

- [1] N. Vaidya, "TCP for Wireless and Mobile Hosts," Tutorial at MobiCom'99, available at <http://www.crhc.uiuc.edu/~nhv/presentations.html>.
- [2] C. Perkins, et al., "Ad hoc On-Demand Distance Vector (AODV) Routing," RFC 3561, Jul. 2003
- [3] 春原他, "アドホックネットワークにおけるTCP通信での性能に関する一考察," 第2回信学ANワークショップ, May 2005.
- [4] 春原他, "バースト誤りのある無線リンクを用いたアドホックネットワークにおけるTCP通信の性能に関する一検討," 信学2005年ソサイエティ大会, Sep. 2005.
- [5] H. Balakrishnan, V. Padmanabhan, S. Seahan, and R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," Proc. of ACM SIGCOMM, Aug. 1996.
- [6] B. Bakshi, P. Krishna, D. Pradhan, and N. Vaidya, "Improving Performance of TCP over Wireless Networks," Proc. of International Conf. Distributed Computing Systems, May 1997.
- [7] Consultative Committee for Space Data Systems (CCSDS), "Space Communications Protocol (SCPS-TP)," Sep. 1997.
- [8] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A Feedback Based Scheme for Improving TCP Performance in Ad-hoc Wireless Networks," Proc. of International Conf. Distributed Computing Systems, 1998.
- [9] D. Kim, C. K. Toh, and Y. Choi, "TCP-Bus: Improving TCP Performance in Wireless Ad hoc Networks," Proc. of ICC 2000, Jun. 2000.
- [10] T. Goff, J. Moronski, and D. Phatak, "Freeze-TCP - A True End-to-End TCP Enhancement Mechanism for Mobile Environments," Proc. of INFOCOM 2000, Mar. 2000.
- [11] S. Floyd, "TCP and Explicit Congestion Notification," ACM Computer Communication Review, Vol.24, pp.10-24, Oct. 1994.
- [12] M. Mathis, J. Mahdavi, S. Floyd and A. Romanow, "TCP Selective Acknowledgment Options," RFC2018, Oct. 1996.